AUTOMATIC ANALYSIS AND ADJUSTMENT OF DIGITAL IMAGES UPON ACQUISITION

COPYRIGHT AUTHORIZATION

TECHNICAL FIELD

The present invention relates to processing of digital images, and more particularly relates to automatically analyzing and adjusting digital images when the images are acquired.

BACKGROUND

Digital images (e.g., images captured by digital cameras or scanners) are generally not perfect photographs. Problems with digital images can include, for example, underexposure, red-eye caused by flash, incorrect orientation and blurriness.

Consumers demand high visual quality and convenience in dealing with digital photographs. Previously, users have relied on third-party imaging software to re-touch digital images. These software packages provide complicated adjustable scalars for users to adjust images manually after the images have been acquired by a PC. While some software packages provide adjustment functions requiring less user expertise or interaction, they normally require the end-user to initiate the adjustment for individual images.

Whatever the benefits of previous techniques, they do not have the advantages of the techniques and tools presented below.

SUMMARY

In summary, the detailed description is directed to various techniques and tools for automatically analyzing and adjusting digital images upon acquisition of the images. Digital images processed using the described techniques and tools can be compressed or uncompressed color or greyscale images in a variety of formats (e.g., JPEG, EXIF, BMP, TIFF, PNB, GIF, WMF, or EMF formats).

In one aspect, an application processes a digital image by analyzing image data from a digital image file; and adjusting the image data (e.g., pixel data) from the digital image file based at least in part on the analysis of the image data. The analyzing and the adjusting are performed automatically upon acquiring the digital image file. The automatic analyzing and adjusting can be turned on or off (e.g., by a user). The image data is acquired from a digital image data source such as a digital camera, scanner, digital video camera or mass-storage device. Acquisition can be initiated at a source location or target location for the digital image, and can be performed via a wired or wireless communication medium locally or over a network. Non-image information (e.g., flash information, focal length, shutter speed, camera model information, aperture setting, or date/time information) also can be analyzed. The adjusting can be based on image characteristic data (e.g., image orientation data, red-eye detection data, blur data, color balance data, exposure data, or noise data). Metadata corresponding to image adjustments can be generated and stored in a digital image file containing an adjusted image. Storing the metadata in the file facilitates preservation of an original version of the image. An image file with adjustment metadata can be provided to a user-mode application that requested the acquisition of the image.

In another aspect, a computer system comprises an image analysis software module for analyzing digital image data at image acquisition time; and an image adjustment software module for adjusting digital image data at image acquisition time based at least in part on the analysis. Features of the system (e.g., the analysis and adjustment modules) can be included in an image acquisition service of an operating system. The system can include an image decoder for decoding compressed digital image data and an image encoder for encoding adjusted digital image data. The image adjustment software module can include one or more processing filters for adjusting the digital image data. The system also can comprise an extensible software architecture operable to allow customization of the image adjustment software module. The architecture can include processing filters encapsulating image adjustment functions. Customizing can include adding, removing or reordering processing filters. The image adjustment software module can be capable of generating metadata corresponding to adjustments of the digital image data, and the system can include a metadata/image integrator for integrating metadata into a file containing an adjusted image.

In another aspect, software for a digital image analysis and adjustment system is developed by receiving a software platform for analyzing and adjusting digital images upon acquisition of the digital images, where the software platform includes a customizable software architecture for adjusting digital image data based on analysis of the digital image data, and creating a custom image adjustment module compatible with the customizable software architecture. The custom image adjustment module adds image adjustment functionality to the system.

In another aspect, a software system providing digital image processing functionality includes a customizable software architecture for adjusting digital image data based on analysis of the digital image data upon acquisition of digital images from a digital image source, wherein the customizable software architecture is capable of operably coupling one or more image adjustment modules encapsulating image adjustment functions to one or more image analysis modules. The functionality of the software system is capable of being customized by altering an arrangement of image adjustment modules operably coupled to the one or more image analysis modules. Altering the arrangement can include adding, removing or changing the functional order of image adjustment modules encapsulating image adjustment functions.

The described techniques and tools can be implemented as features of an operating system environment. In some embodiments, the described techniques and tools are implemented in a managed code environment or in a background service of an operating system environment.

The various techniques and tools can be used in combination or independently. Different embodiments implement one or more of the described techniques and tools.

Additional features and advantages will be made apparent from the following detailed description of different embodiments that proceeds with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a suitable computing environment for implementing tools for automatic analysis and adjustment of digital images.

Figure 2 is a block diagram illustrating a pull model of image acquisition.

Figure 3 is a block diagram illustrating a push model of image acquisition.

Figure 4 is a flow diagram showing an acquisition-triggered automatic analysis and adjustment technique for digital images.

Figure 5 is a flow diagram showing an acquisition-triggered automatic analysis and adjustment technique for digital images.

Figure 6 is a block diagram of a computer system including automatic image analysis and adjustment tools implemented in an image acquisition service.

Figure 7 is a block diagram of an automatic image analysis and adjustment tool that interacts with an image capture device.

Figure 8 is a block diagram of an image file with adjustment metadata.

Figure 9 is a detailed block diagram of an automatic image analysis and adjustment tool that provides image files with adjustment metadata to a user mode application.

DETAILED DESCRIPTION

Techniques and tools for automatic analysis and adjustment of digital images are described. The techniques and tools facilitate digital image analysis and adjustment functions triggered by image acquisition. The digital images can be color, grey-scale, or other types of images in a variety of formats (e.g., JPEG, EXIF, BMP, TIFF, PNB, GIF, WMF and/or EMF formats). The digital images can be acquired from any device capable of capturing or storing digital still images (e.g., digital cameras, scanners, digital camcorders, webcams, mass storage devices, etc.).

For example, using one or more of the described techniques and tools, an application automatically adjusts digital photographs transferred to a target location (e.g., a PC) from a source location (e.g., a digital camera). The described techniques and tools save users from the burden of manually adjusting digital images after they are acquired to the PC.

Features of some embodiments of the described techniques and tools include: (1) image adjustment without user interaction based on image analysis results; (2) batch processing of images; and (3) custom configuration and updating of image analysis and adjustment processing. Automatic image analysis and adjustment can be turned on and off (e.g., by a user via interaction with a graphical user interface).

Some embodiments facilitate preservation of original versions of digital images by generating metadata corresponding to adjustments of the digital image data. The metadata describes the adjustments and allows the adjustments to be reversed or "undone."

Some embodiments include an extensible architecture. The extensible architecture allows independent hardware vendors (IHVs) and independent software vendors (ISVs) to advocate their technologies by providing their own "plug-ins" to perform image analysis and/or adjustment. For example, filters for performing image adjustments can be added, removed, or reordered to change how images are adjusted in response to image analysis.

Some embodiments are included in a subsystem (e.g., an image acquisition service) within an operating system. An image acquisition service processes requests (e.g., from user-mode applications via an application programming interface) to acquire image files, and communicates with devices at a source location (e.g., via drivers such as user mode drivers or kernel mode drivers) to transfer the image files.

Some embodiments operate in a managed code environment, with automatic image analysis and adjustment tools and techniques being outside an image acquisition service. Automatic image analysis and adjustment tools and techniques also can be implemented in a background service in an operating system.

I. Computing Environment

Figure 1 illustrates a generalized example of a suitable computing environment 100 in which the described techniques and tools may be implemented. The computing environment 100 is not intended to suggest any limitation as to scope of use or functionality of the invention, as the present invention may be implemented in diverse general-purpose or special-purpose computing environments.

With reference to Figure 1, the computing environment 100 includes at least one processing unit 110 and memory 120. In Figure 1, this most basic configuration 130 is included within a dashed line. The processing unit 110 executes computer-executable instructions and may be a real or a virtual processor. In a multi-processing system, multiple processing units execute computer-executable instructions to increase processing power. The memory 120 may be volatile memory (e.g., registers, cache, RAM), non-volatile memory (e.g., ROM, EEPROM, flash memory, etc.), or some combination of the two. The memory 120 stores software 180 implementing tools for automatic analysis and adjustment of digital images.

A computing environment may have additional features. For example, the computing environment 100 includes storage 140, one or more input devices 150, one or more output devices 160, and one or more communication connections 170. An interconnection mechanism (not shown) such as a bus, controller, or network interconnects the components of the computing environment 100. Typically, operating system software (not shown) provides an operating environment for other software executing in the computing environment 100, and coordinates activities of the components of the computing environment 100. In some embodiments, operating system software includes tools for automatic analysis and adjustment of digital images.

The computing environment desirably includes digital image processing hardware (e.g., a scanning device and/or digital camera) and software.

The storage 140 may be removable or non-removable, and includes magnetic disks, magnetic tapes or cassettes, CD-ROMs, CD-RWs, DVDs, or any other medium which can be used to store information and which can be accessed within the computing environment 100. The storage 140 stores instructions for the software 180 implementing tools for automatic analysis and adjustment of digital images.

The input device(s) 150 may be a touch input device such as a keyboard, mouse, pen, or trackball, a voice input device, a scanning device, a digital camera, or another device that provides input to the computing environment 100. The output device(s) 160 may be a display, printer, speaker, or another device that provides output from the computing environment 100.

The communication connection(s) 170 enable communication over a communication medium to another computing entity. The communication medium conveys information such as computer-executable instructions, digital image information, audio or video information, or other data in a modulated data signal. A modulated data signal is a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media include wired or wireless techniques implemented with an electrical, optical, RF, infrared, acoustic, or other carrier.

The invention can be described in the general context of computer-readable media. Computer-readable media are any available media that can be accessed within a computing environment. By way of example, and not limitation, with the computing environment 100, computer-readable media include memory 120, storage 140, communication media, and combinations of any of the above.

The invention can be described in the general context of computer-executable instructions, such as those included in program modules, being executed in a computing environment on a target real or virtual processor. Generally, program modules include routines, programs, libraries, objects, classes, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The functionality of the program modules may be combined or split between program modules as desired in various embodiments. Computer-executable instructions for program modules may be executed within a local or distributed computing environment.

For the sake of presentation, the detailed description uses terms like "analyze," "determine," "adjust," and "acquire" to describe computer operations in a computing environment. These terms are high-level abstractions for operations performed by a computer, and should not be confused with acts performed by a human being. The actual computer operations corresponding to these terms vary depending on implementation.

II. <u>Acquisition of Digital Images</u>

Image acquisition refers to activities associated with transferring one or more digital images from a source location to a target location. A source location is a device that 1) is capable of storing or capturing one or more digital images, and 2) allows one or more digital images to be transferred to a target location. A target location is a device that 1) is capable of storing one or more digital images, and 2) is capable of receiving digital images from a source location. For example, a PC can acquire images from a source location (e.g., a digital camera, scanner, or other device) connected to the PC via a wired connection (e.g., via a serial port or USB connection) or a wireless connection.

Referring to Figure 2, in a "pull" model 200 of image acquisition, a target location 210 initiates (e.g., by user interaction with an application at the target location) image acquisition of a digital image 220 from a source location 230. Target location 210 sends a request 240 to the source location 230 and receives the digital image 220 from the source location in response to the request. For example, an application at target location 210 (e.g., a desktop PC, handheld computer, or other target location) locates and selects a digital image to acquire. In one common scenario, a PC user instructs a PC to acquire a digital image from a digital camera connected to the PC by interacting with the PC via a graphical user interface ("GUI") for an application on the

PC. For example, a PC user can interact with a GUI for a device "wizard," an imaging application (e.g., Microsoft® Picture It! or Photoshop by Adobe Systems Inc.) or an image acquisition service in an operating system.

Referring to Figure 3, in a "push" model 300 of image acquisition, a source location 330 (e.g., a digital camera) initiates image acquisition by sending device event data 350 to target location 310 (e.g., a PC). In response, target location 310 requests acquisition of a digital image 320 by sending a request 340 to the source location 330. For example, in one common scenario, a connection of a scanner to a PC produces a device event, and in response, an application launches on the PC and requests retrieval of a scanned image from the connected scanner. A source location can also send a message (e.g., a notification that a digital image is now available for retrieval) to an application that is already running at the target location. In response, the application requests retrieval of the image. For example, a scanner connected to a PC notifies a digital image editor running on the PC that a scanned image is available for retrieval. A user can also cause the digital image source to initiate the transfer (e.g., by pushing a button on a digital camera, scanner, or video camera)

III. Automatic Image Analysis and Adjustment

Described embodiments include techniques and tools facilitating automatic analysis and adjustment of digital images initiated by image acquisition.

The features of these techniques and tools include:

1. Automatic adjustment of digital images based on image analysis results; no user interaction is needed, although a user can have the option of turning the automatic analysis and adjustment on or off.
2. Automatic adjustment of digital images based on image analysis results in batch mode, when acquisition is done in batch mode.
3. An extensible architecture allowing image analysis and adjustment tools and techniques to be custom-configured and updated.

Figure 4 shows a technique 400 for automatically analyzing and adjusting digital images. First, at 410, image acquisition begins. Image acquisition may be according to a push model or a pull model of image acquisition, examples of which are described above. Image acquisition initiates automatic analysis and adjustment of the image at 420.

In some embodiments, the process of automatically analyzing and adjusting digital images includes three main stages: analyzing the image; processing the image data to render new image data based on the image analysis results; and generating and integrating new metadata into the image data file. Figure 5 shows a technique 500 that includes these three main stages (represented in Figure 5 at blocks 540, 550 and 560).

First, when image acquisition is initiated, the automatic analysis and adjustment service receives a request to acquire an image data file at 510. At 520, the automatic image analysis and adjustment service passes this request to an external interface. For example, referring to Figure 6, automatic image analysis and adjustment tools 610 use one or more interfaces to pass the request to image capture device 680 via hardware interface 670.

Referring again to Figure 5, at 530 the automatic image analysis and adjustment service receives an image data file (e.g., from an image capture device). The automatic image analysis and adjustment service analyzes the image data at 540 and adjusts the image data based on the analysis at 550. The automatic image analysis and adjustment service generates metadata corresponding to the image data adjustment and integrates the new metadata into the image data file at 560. The automatic image analysis and adjustment service provides the image data file with the new metadata to the entity (e.g., user mode process 630 (Figure 6)) that requested the acquisition of the image data file.

Figure 6 shows a block diagram of an exemplary computer system 600 including an automatic image analysis and adjustment service 610. In arrangements such as the one shown in Figure 6, an automatic image analysis and adjustment service works within an image acquisition service 620. A user mode process 630 at a target location interacts with the image acquisition service and automatic image analysis and adjustment tools 610 via one or more interfaces (e.g., application programming interfaces (APIs) 640) to acquire an image. Image acquisition service 620 allows user mode process 630 to acquire digital images from source locations such as image capture device 680 by facilitating communication with user mode drivers 650, kernel mode drivers 660, and hardware interface 670.

An image acquisition service is a subsystem within an operating system that processes requests from user-mode applications and communicates with device drivers (e.g., camera or scanner drivers) to acquire image files. For example, in some embodiments, the automatic image analysis and adjustment tools are included in a Microsoft® Windows® Image Acquisition (WIA) architecture. In such embodiments, the automatic image analysis and adjustment service tools can leverage features of WIA (e.g., GUIs, broad device coverage) while saving the user from the burden of adjusting digital images after they are acquired to the PC.

WIA includes an API called *IWiaDataTransfer*. Programming interfaces such as *IWiaDataTransfer* can be used in conjunction with an automatic image analysis and adjustment service. The API can be modified to include flags signaling whether and in what manner automatic image analysis and adjustment is performed. *IWiaDataTransfer* includes the member function *IWiaDataTransfer::idtGetData*, which is called to retrieve files from an appropriate device. For example, in some cases, images are retrieved from a device of a particular class (e.g., "Image Acquisition Device"). The device class can be detected based on a plug-and-play globally-unique identifier ("GUID") stored in an INF file that indicates to WIA that the device is an image capture device, a USB mass-storage device, etc. A software component called "Autoplay" can be used to detect whether retrievable pictures are stored on the device.

In addition to using image acquisition service programming interfaces or modified image acquisition service programming interfaces, the automatic image analysis and adjustment service can use custom programming interfaces created specifically for the service. In some embodiments, the automatic image analysis and adjustment service includes programming interfaces such as *IEnumFilters* and *IImageAutoProcess*. Details for the *IEnumFilters* and *IImageAutoProcess* APIs are provided below:

```
//
// IEnumFilters interface declarations
//

DECLARE_INTERFACE_(IEnumFilters , IUnknown)
{
   STDMETHOD(Init)(DWORD dwFlags);
   STDMETHOD(Next)(ULONG cFilters, IImageFilter **ppFilter, ULONG
*pcFeteched);
   STDMETHOD(Skip)(ULONG cFilters);
   STDMETHOD(Reset)(void);
   STDMETHOD(Clone)(IEnumFilters **ppEnum);
};

//
// IImageAutoProcess interface declarations
//

DECLARE_INTERFACE_(IImageAutoProcess, IUnknown)
{
   STDMETHOD(Initialize)(TCHAR *tcsFilename, DWORD dwFlags);
   STDMETHOD(Initialize)(IStream *pStream, DWORD dwFlags);
   STDMETHOD(Initialize)(BYTE *pBuffer, DWORD dwFlags);
   STDMETHOD(EnumFilters)(IEnumFilters **ppEnum);
   STDMETHOD(Process)(void);
   STDMETHOD(Finalize)(void);
};
```

Alternatively, the automatic image analysis and adjustment service can use other custom or pre-existing interfaces for performing image analysis and adjustment.

Figure 7 shows an example of an automatic image analysis and adjustment service 700 used in some embodiments. The automatic image analysis and adjustment service 700 receives image files (e.g., image file 710) transferred from source locations such as image capture device 720. Image analysis module 730 analyzes image data 740 from image file 710. Image analysis module 730 analyzes the image data 740 to identify ways in which the image can be adjusted to improve the quality of the image. For example, image analysis module 730 can perform red-eye detection, blur detection, orientation analysis, noise detection, color balance analysis, exposure analysis, and/or other types of image analysis. Image adjustment module 750 then adjusts the image data, if necessary, based on the analysis, and generates adjustment metadata 760. Metadata/image integrator 770 integrates the adjustment metadata 760 and outputs image file with adjustment metadata 780.

For example, referring to Figure 8, an image file 800 includes a header 810, metadata section 820, and data bits section 830. Adjustment metadata 840 is stored in the metadata section 820. Alternatively, data within the image file 800 can be arranged in other ways.

Referring again to Figure 7, in some embodiments, metadata/image integrator 770 adds the generated metadata to image file 710. Alternatively, metadata/image integrator 770 adds the generated metadata to another file (e.g., a copy or another version of image file 710).

Figure 9 shows another example of an automatic image analysis and adjustment service 900 used in some embodiments. An image file reader 910 in the automatic image analysis and adjustment service 900 reads image files (e.g., image file 915) received from source locations such as an image capture device (not shown). Image file reader 910 can be any entity capable of reading an image file. For example, for image files in JPEG format, the image reader may be a JPEG image file reader. Image file reader 910 can also be an image file reader capable of reading image files in other formats, such as EXIF, BMP, TIFF, PNB, GIF, WMF and/or EMF formats. Automatic image analysis and adjustment service 900 can incorporate a single image file reader capable of reading one or more file formats, or multiple image file readers.

From image file reader 910 comes image data 920 and non-image information 925. Non-image information can include data indicating, for example, whether a flash was used in a photograph, focal length, shutter speed, camera model, aperture setting, date, etc. Non-image information can be used in several ways. For example, the automatic image analysis and adjustment service 900 may use an indicator of whether a flash was used in a digital image to determine whether to check for red-eye when analyzing the image, or whether to use a particular filter to remove red-eye from an image.

Image data 920 goes to decoder 930 for decoding. Because image analysis is typically done in the raw image data space, decoder 930 can be used to decode the image data 920 into raw image buffer 935. Decoder 930 can be any entity capable of decoding a compressed image. For example, to provide capability for decoding image files in JPEG format, decoder 930 can be a JPEG decoder. Automatic image analysis and adjustment service 900 can incorporate a single decoder capable of reading one or more file formats, or multiple decoders. If no image decoding is needed, the decoder can be bypassed or omitted.

Raw image buffer 935 receives decoded raw image data. Image analysis module 940 analyzes the decoded raw image data (e.g., by performing red-eye detection, blur detection, orientation analysis, noise detection, color balance analysis, exposure analysis, and/or other types of image analysis). Image analysis module 940 then sends pixel data 945 and image characteristics (i.e., image analysis results) 950 to other components of automatic image analysis and adjustment service 900.

Automatic image analysis and adjustment service includes one or more filters for adjusting image data. These processing filters cascade together to perform image adjustment processing on pixel data for an image based on the image analysis results. For example, automatic image analysis and adjustment service 900 includes filters 955, 960, 965 and 970 for adjusting image data based on analysis of the image data. The automatic image analysis and adjustment service uses filters to adjust image data to compensate for red-eye, blurriness, incorrect orientation, noise, incorrect color balance, incorrect exposure, and/or other types of image problems. While some embodiments include several filters, the filters need not all be used when analyzing and adjusting images. Filters can be enabled or disabled (either automatically or by user interaction) in order to process images in different ways. For example, when analyzing/adjusting a digital photograph where no flash was used, a user can disable (or the service can automatically disable) the red-eye filter.

The filters generate adjustment metadata 975 corresponding to adjustments of the pixel data. At the end of the processing chain, metadata/image integrator 980 updates and integrates metadata for the image into an output file 985 which contains the adjusted pixel data. In some embodiments, metadata/image integrator 980 adds the generated metadata to image file 915. Alternatively, metadata/image integrator

980 adds the generated metadata to another file (e.g., a copy of or additional version of the image file 915). The automatic image analysis and adjustment service can also include an encoder (not shown) to re-encode images that have been previously decoded.

While automatic image analysis and adjustment service filters can be chained together, the filters are operable as individual processing units. Each filter performs a specific processing function. The modularity of the filters allows new filters to be added and allows existing filters to be chained together in different arrangements, omitted, or replaced with different filters. This architecture facilitates extensibility and flexibility, allowing third parties, such as independent hardware vendors (IHVs) or independent software vendors (ISVs), to provide additional and/or improved digital image adjustment functionality to the service. New filters (e.g., filters provided by IHVs or ISVs) can be authenticated and/or approved before installation into the service.

In some embodiments, the automatic image analysis and adjustment service uses the following flags to define the scope of image analysis/adjustment filters to enumerate:

```
enum ImageFilterSwitches
{
   IMAGE_FILTER_SWITCH_ALL
   IMAGE_FILTER_SWITCH_AUTHENTICATED
   IMAGE_FILTER_SWITCH_USER_SELECT
   IMAGE_FILTER_SWITCH_INBOX
   IMAGE_FILTER_SWITCH_VENDOR
};   // Flags used to define the scope of filters to enumerate
```

Alternatively, the service uses different flags or other ways of defining the scope of filters to enumerate.

A. **Example: Correcting the Orientation of an EXIF Imag**

Suppose the automatic image analysis and adjustment processing chain comprises auto-orientation: automatically detecting image orientation and rotating it to the correct direction. The image analysis module 940 will first analyze the image textures and color distribution of the image contained in input EXIF image file to decide on the correct orientation for the image. Then, one or more filters process the image to rotate it, if necessary, to the correct orientation. Finally, the EXIF tag for the orientation of the image is created or updated to reflect that the image has been rotated to the correct orientation. For more information on the EXIF file format and EXIF tags, see JEITA CP-3451, *Exchangeable Image File Format for Digital Still Cameras: Exif Version 2.2* (April 2002).

B. Implementation as Operating System Features

The tools and techniques described above can be implemented as operating system features. With an operating system that includes one or more of the described tools and techniques, users can take advantage of automatic image analysis and adjustment without using additional applications.

Some embodiments operate in a managed code environment. For example, automatic image analysis and adjustment tools and techniques are implemented within a managed-code operating system, but outside an image acquisition service, to facilitate modifications or updates to the tools.

Automatic image analysis and adjustment tools and techniques also can be implemented in a background service of an operating system.

Having described and illustrated the principles of our invention with reference to described embodiments, it will be recognized that the described embodiments can be modified in arrangement and detail without departing from such principles. It should be understood that the programs, processes, or methods described herein are not related or limited to any particular type of computing environment, unless indicated otherwise. Various types of general purpose or specialized computing environments may be used with or perform operations in accordance with the teachings described herein. Elements of the described embodiments shown in software may be implemented in hardware and vice versa.

In view of the many possible embodiments to which the principles of our invention may be applied, we claim as our invention all such embodiments as may come within the scope and spirit of the following claims and equivalents thereto.